AI Magazine Volume 13 Number 4 (1992) (© AAAI)

The AI Program at the National Aeronautics & Space Administration

Lessons Learned During the First Seven Years

Melvin D. Montemerlo

■ Thsi article is a slightly modified version of an invited address that was given at the Eighth IEEE Conference on Artificial Intelligence for Applications in Monterey, California, on 2 March 1992. It describes the lessons learned in developing and implementing the Artificial Intelligence Research and Development Program at the National Aeronautics and Space Administration (NASA). In so doing, the article provides a historical perspective of the program in terms of the stages it went through as it matured. These stages are similar to the "ages of artificial intelligence" that Pat Winston described a year before the NASA program was initiated. The final section of the article attempts to generalize some of the lessons learned during the first seven years of the NASA AI program into AI program management heuristics.

This article is about the lessons learned in setting up and carrying out the first seven years of the Artificial Intelligence Research and Development Program at the National Aeronautics and Space Administration (NASA). This AI program is sponsored by NASA's Office of Aeronautics and Space Technology. The program conducts research and development at the NASA centers (Ames, Lewis, Marshall, Kennedy, Johnson, Goddard, and the Jet Propulsion Laboratory [JPL]). It also sponsors research in academia and industry, primarily through Ames Research Center, which is the lead center for AI research at NASA. The AI group at Ames, which is headed by Peter Friedland, has particular strengths in the areas of planning and scheduling, learning, and reasoning about physical systems. The teams at the other centers are primarily devoted to applying AI technologies within their centers.

As described later in the article, the program has had a number of successes in both developing and applying AI technology within NASA. NASA's AI program has implemented AI applications at Johnson Space Center (JSC), Kennedy Space Center (KSC), and JPL that have revolutionized NASA's approach to ground control of manned and unmanned missions. It has also developed AI-based data analysis tools that are in operational use within NASA and that have been distributed widely outside NASA.

However, the purpose of this article is not to list the accomplishments of the program. Rather, it is to attempt to describe the lessons learned in the process of putting the program together and carrying it out. How was the program sold originally? How was it planned? Did the plan work? If not, why not? How did the program readjust? What pitfalls were faced, and how would they be handled differently now? What are the heuristics used to keep NASA's AI ship afloat in the churning seas of government politics?

Although the AI program management team never got lost in the process of setting up the AI program, there were a few times when it was temporarily directionally disoriented. There were encounters with the unforeseen that called for real-time reactive replanning. In this article, I try to pinpoint the important events in NASA's AI history and to encapsulate the wisdom that was garnered in the process. As you see, a lot of the lessons

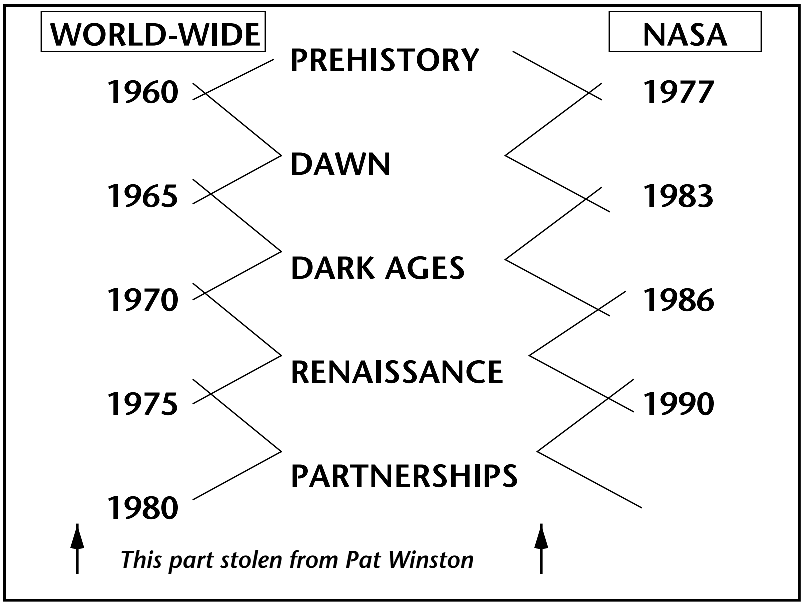


Figure 1. The Historical Periods of AI.

Pat Winston wrote about the "ages of AI" in 1984. Winston's dates for each age appear in the left column. The dates on the right are when NASA's AI program went through the same stages of development.

that were learned had already been learned by others. This article is an attempt to spread the word a little further.

A Historical Perspective

NASA has always been a leader in automation. All NASA's unmanned missions have been wonders of conventional automation—Pioneer, Viking, Voyager, and so on. Even the manned missions—Mercury, Gemini, Apollo, and Shuttle—have vast amounts of automation. In going from earth to orbit, the shuttle is almost fully autonomous. There are only a few points at which humans can influence its path if problems are encountered. However, the new generation of automation, with its underpinnings in AI, is relatively new at NASA.

I searched for a way to organize the major events in NASA's AI history. In the process, I ran across Pat Winston's (1984) description of the "ages of artificial intelligence." He subdivides the history of AI into five eras: Prehistory, Dawn, Dark Ages, Renaissance, and Partnerships. These eras can be seen in figure 1. The happenings at NASA fit naturally into this same pattern. As I reread his paper, I had the feeling that Winston was being prescient as well as reflective. He foretold what would happen at NASA. Let's take a look.

Prehistory

For Winston, the Prehistory of AI was the era before researchers had the computers to implement computational approaches to intelligence. There were those who had the vision but not the tools. He has this era ending about 1960. For NASA, Prehistory was the era to about 1977. NASA had the computers at the time but not the people with the vision. Both are necessary to make it happen.

Dawn

For Winston, the Dawn of AI was the 1960–1965 era, when both the computers and the vision were there, but the vision was romantic. Predictions were made that in 10 years, computers would be as smart as people. These were the predictions of conscientious scientists, who were preparing the way for things to come. During this time, there were some successes, such as MACSYMA.

For NASA, the Dawn of AI occurred in the 1977–1983 time frame, when those with the vision began to emerge. In 1977, NASA's research code brought together a distinguished group of advisers to consult with NASA on how the field of computer science, especially the areas of AI and robotics, could help NASA perform its mission. The group was headed by Carl Sagan. The membership list of the committee is impressive: Carl Sagan, chairperson; Elliot Levinthal; Raj Reddy; Jack Minker; James Albus; Marvin Minsky; Robert Balzer; Charles Rieger; Thomas Binford; Donald Norman; Ralph Gonzales; Thomas Sheridan; Peter Hart; Patrick Winston; B. Gentry Lee; and Stephan Yerazunis.

Between June 1977 and December 1978, this group spent an incredible 2500 person-hours on the study. The conclusions of the committee were as follows (JPL 1980): First, NASA is 5 to 15 years behind the leading edge in computer science and technology. Second, technology decisions are, to much too great a degree, dictated by specific mission goals, powerfully impeding NASA's use of modern computer science and technology. Third, the overall importance of machine intelligence and robotics for NASA has not widely been appreciated within the agency, and NASA has made no serious effort to attract bright, young scientists in these fields. Fourth, the advances in machine intelligence and robotics needed to make future space missions economical and feasible will not happen without a major long-term commitment and centralized coordinated support.

As you can tell from these recommendations, the committee pulled no punches. Its final report, which was published in 1980, still makes good reading today. This activity led to the formation of NASA's Computer Science Research Program in 1982, which began funding research in AI at JPL. Prior to this time, there was one researcher at JPL working in the area of AI. It was Steve Vere, and he was developing DEVISER, a scheduler for telemetry uplink.

In 1984, the AI dream at NASA got a jump start. Congress passed a bill that initiated the funding of the Space Station program. This bill contained strong words about automation and robotics for the space station (figure 2). It asked NASA to identify specific space station systems that would advance automation and robotics technologies, the development of which should be estimated to cost no less than 10 percent of the total space station cost, which at that time was to be $8 billion.

The bill gave rise to NASA's Artificial Intelligence Research and Development Program, which was initiated in 1985, with funding of $4 million. I have the privilege of having helped put this program together and of having been program manager since its inception. NASA Ames Research Center was named the lead center. JPL, KSC, JSC, Langley, Marshall, Lewis, and Goddard were the participating centers.

The bill also gave rise to two advisory committees, one made up of NASA insiders and one made up of experts from academia and industry. Their job was to advise NASA on what it should do to develop and apply AI and robotics technology to the space station effort. The outside advisory group recommended that $100 to $190 million each year be fenced for advanced research and development in automation and robotics and that NASA's administrator have an assistant (who was jokingly referred to as "a 2000-pound gorilla") to ensure that automation and robotics is well taken care of and not shunted to the side. We never got this level of funding or the 2000-pound gorilla, but NASA's AI research program grew from $4 million each year in 1985 to $13 million in 1988, and it has stayed at about this level ever since. The program was aimed not only at the space station but also at the shuttle and space science. It is this program that this article is about.

LEGISLATION ENABLING SPACE STATION

PUBLIC LAW 98-371 OF THE 98TH CONGRESS, DATED JULY 18, 1984, NATIONAL AERONAUTICS AND SPACE ADMINISTRATION RESEARCH AND DEVELOPMENT, 98 STAT. 1227, RESEARCH AND PROGRAM MANAGEMENT REPORTS states:

"Provided further, that the Administrator shall establish an Advanced Technology Advisory Committee in conjunction with NASA's Space Station program and that the Committee shall prepare a report by April 1, 1985,

identifying specific space station systems which advance automation and robotics technologies, not in use in existing spacecraft, and

that the development of such systems shall be estimated to cost no less than 10 per centum of the total space station cost."

Figure 2. Legislation Enabling the Space Station.

The public law that enabled Space Station Freedom specifically focused on automation and robotics. This public law resulted in a number of developments at the National Aeronautics and Space Administration, including the establishment of an AI research program.

The Dark Ages

Winston placed the Dark Ages of AI from about 1965 to about 1970. This was an era in which little happened. He attributed this dry spell to the feeling that creating intelligent systems would be as easy as was predicted during the Dawn era. It was an era filled with romantic, simplistic notions of what AI would be able to do in a decade. Winston said that it was a time of looking for a kind of *philosopher's stone*, a device that when put in the computer would make it intelligent. The Dark Ages of AI at NASA were from about 1984 to 1986. Figure 3 was the most often used AI vugraph at NASA during this period. The implication was that all the program had to do to make systems intelligent was to fill in the boxes. Figures 4 and 5 show how it was to be done. Obviously, the program's philosopher's stone didn't work either. No one could figure out how to fill in all the boxes.

During 1985 and 1986, proposals seemed to come out of the woodwork. Most of NASA's prospective contractors put together AI groups. The bywords of the time were expert systems and full autonomy. It was predicted that expert systems would exist throughout the space station, and they would work together in planning, scheduling, process control, monitoring, fault detection and diagnosis, and so on. These were heady days. Of course, there were few trained and experienced AI researchers with NASA at the time.

JSC hosted a week-long workshop in April 1985 in which each NASA center got to describe the AI and robotics projects that it had under way. By the end of the second day, the total number of expert systems that had been described as having been developed was about 100. However, none of these expert systems ever became operational.

NASA's AI research program was not immune from overselling and overpredicting. The objec-

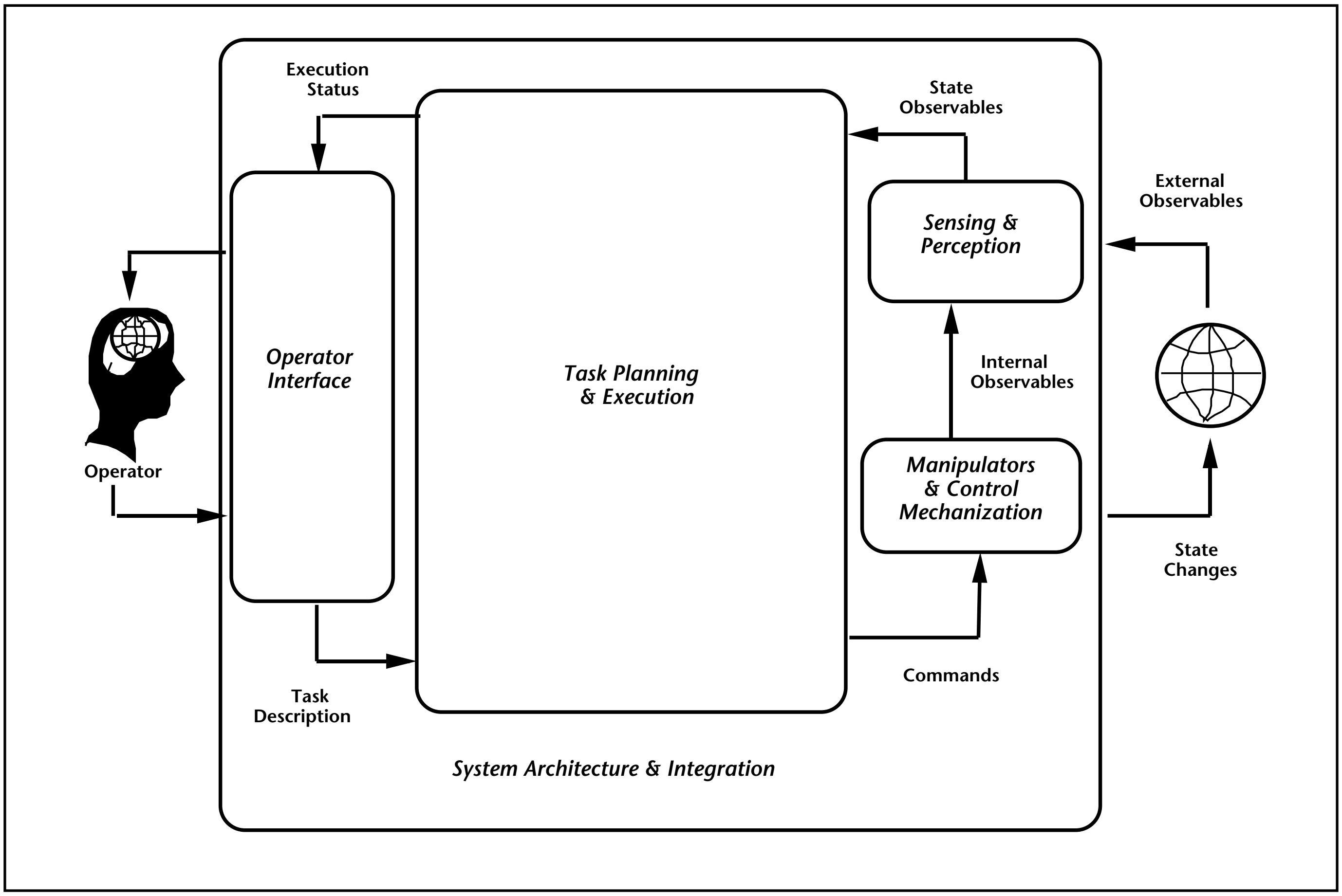


Figure 3. The Original Architecture for an Automated System.

This figure, which was produced in 1984, was widely used at the National Aeronautics and Space Administration to describe what it takes to make an intelligent automated system.

tives stated for the program during its inception were to decrease manpower-intensive tasks in ground mission operations by 75 percent, use real-time expert systems with accompanying productivity gains of 25 percent, and decrease the documentation required for troubleshooting and diagnostics by 80 percent. These objectives were written down in a white paper (Montemerlo and Holcomb 1986) that described the rationale, content, and plans of the newly developed program. These goals were to be achieved by 1995!

The automation and robotics white paper also contained a sequence of planned demonstrations for the AI program (figure 6). The background on the development of this sequence is as follows: Back in 1984, I was asked to put together a 10-year plan for the AI program that would be modeled on the Autonomous Land Vehicle Program of the Defense Advanced Research Projects Agency. It would contain a sequence of ever-more-capable demonstrations, and it would be coupled closely with a fundamental research program. The predicted results of the fundamental research program were scheduled and planned to be fed into the demonstration sequence.

The first demonstration was a success. It consisted of automated monitoring and control of a thermal subsystem test bed in nominal and off-nominal conditions. However, it was quickly realized that the second proposed demonstration couldn't be done. The intelligent autonomous control of two complex subsystems simultaneously was too difficult for the technology of the time.

As Winston (1984) said, "The Dark Age was largely fueled by over-expectation" (p. 4). It is easy to look back and criticize the original predictions of the NASA AI program. Hindsight is 20/20. If the NASA AI program had had the people, the technology, and the experience that it has now, the predictions would have been more realistic.

Renaissance

For Winston, the Renaissance of AI was a

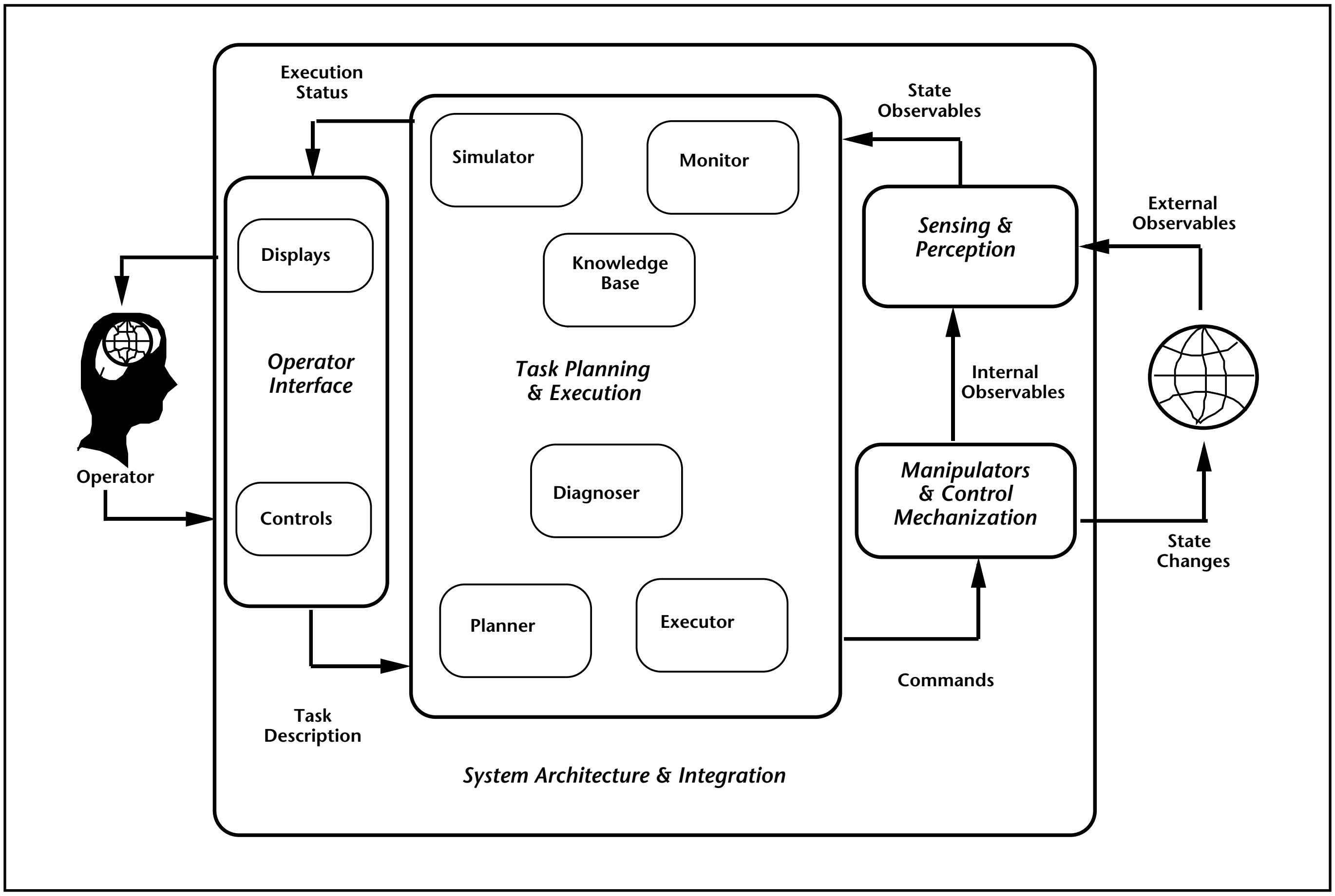


Figure 4. The Revised Architecture for an Automated System.

The original vugraph engendered as many questions as it answered, so this one was developed to explain what the first one meant. It provided more detail.

period in which AI researchers focused more on making systems that worked and that caught people's attention. He placed this period from about 1970 to 1975, when systems such as MYCIN were born. The Renaissance era for AI in NASA took place from 1987 to 1990. It was a time when expectations were revised from revolution to evolution.

NASA AI researchers realized that it is hard enough to transfer established AI technology into NASA projects. They found that project managers are, by nature, conservative people. In putting together large space projects, it is important not to fail. The scientists who previously put together expert systems in the laboratory and then showed them to potential users learned about rejection.

Part of the reason that NASA didn't just skip the Dark Ages of AI was that in 1985, when money became available to start the AI program, precious little AI expertise was on board at NASA.

In 1986, NASA Ames Research Center, the lead center for AI, hired Peter Friedland. Friedland was an established member of the AI community, and he was able to attract bright, capable people as principal investigators. They, in turn, brought in still others. JPL increased its AI staff. The other centers also did some hiring and some training.

During the Dark Ages of AI at NASA, one didn't propose to do simple, useful things. One proposed grandiose projects. Friedland and I had been bucking the tide by advocating the idea that the program needed to make small, early "wins" by applying proven AI technology while they work on enhancing existing technology through fundamental research. This philosophy met with limited acceptance at first. The old weltanschauung didn't die easily. I remember the point in time when it turned around.

Friedland gathered a group of experienced AI professionals, consisting of Brad Allen, Bruce Bullock, Jaime Carbonell, Bob Englemore, David Mishelevich, and Ben Wah. The team, sponsored by NASA's Office of Space Station, went to all the NASA centers to deter-

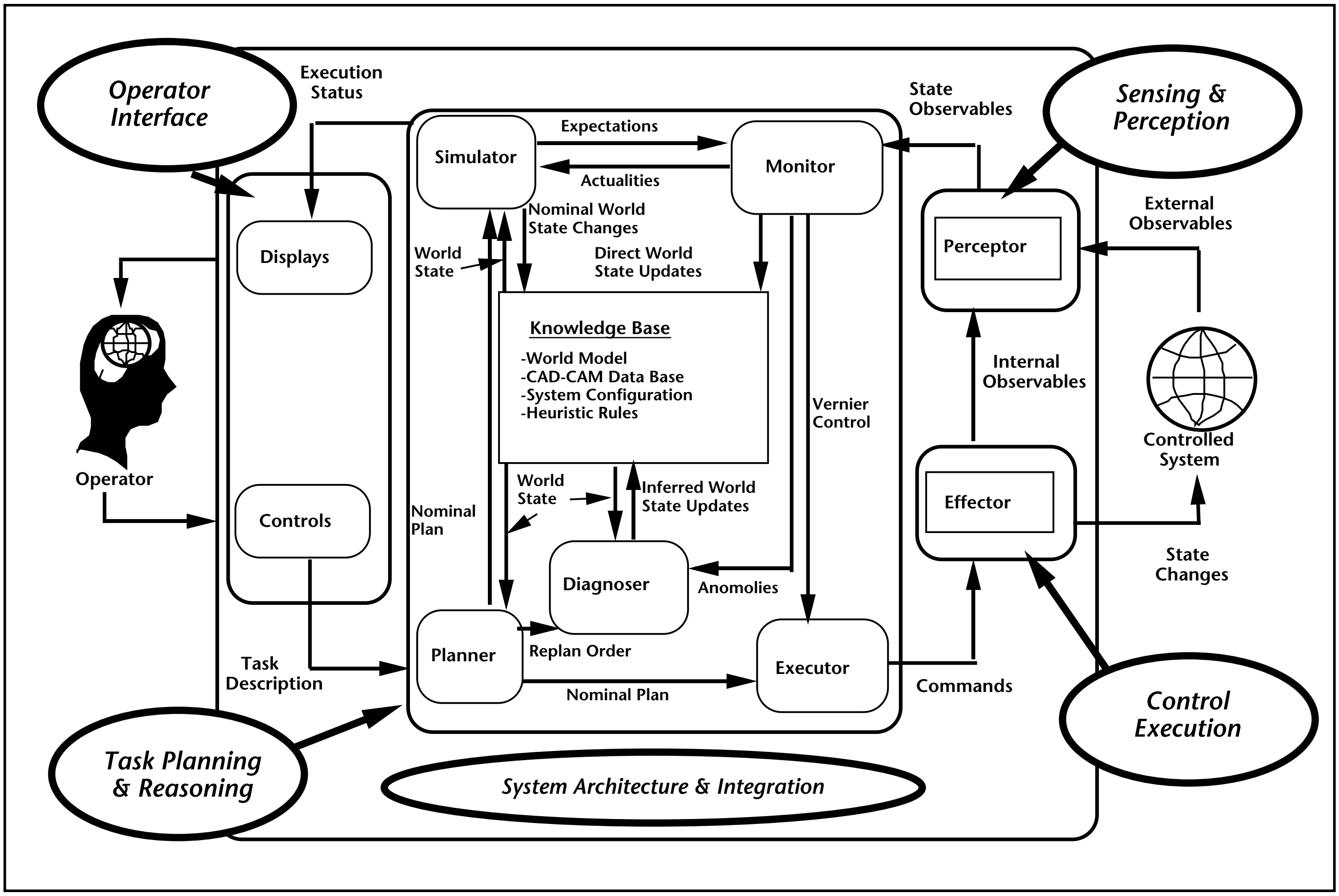


Figure 5. The Final Architecture for an Automated System.

The escalation in detail and complexity of the architecture for an automated system came to an end with this chart. It was both too complex a chart to use for briefings and too unwieldy a formulation to guide a research program. At this point, the research program gave up the idea of developing an overall structure for intelligent machines and focused on developing and applying technologies for more specific tasks (planning, scheduling, monitoring, diagnosis, learning, data analysis, and so on).

mine what potential applications of knowledge-based systems would be most valuable in support of Space Station *Freedom*. On 17 June 1988, Friedland made a presentation of these findings to two associate administrators at NASA—the one for the space station and the one for research (NASA 1988). The recommendations were to (1) capitalize on existing NASA expertise; (2) not force development environments to be the same as delivery environments; (3) make sure users are involved from the start; (4) begin systems as advisory, and plan for in-the-loop control; (5) deal with real-time data and control issues early; and (6) quickly develop a reasonable validation and verification standard.

At the same meeting, Ed Feigenbaum made a presentation that emphasized the approach of "cherry picking the easy one's first." One of NASA's learned lessons is, therefore, Ed's edict: Cherry pick the easy ones first. After this set of presentations, Ed's edict and the six recommendations of the group were legitimized at NASA, and the pace of AI progress accelerated! Feigenbaum, together with the members of Friedland's group, had a first-order effect on enabling the success of NASA's AI program.

The first cherries that AI program members decided to pick were in the existing NASA Mission Control Centers (MCCs). The goal was to implement established AI techniques in these centers. Three such tasks were initiated:

The first task was a rule-based expert system for Shuttle Mission Control at JSC. The application was called INCO (integrated communications officer). INCO is also the name of the console in the Shuttle Mission Control room at which the integrated communications officer works. John Muratore was the integrated communications officer at the time. He was also the person who headed the development of the INCO AI application.

The second task was a rule-based expert

1988 Automated Control of Mission Operations Subsystem ("Intelligent Aide")	1990 Automated Control of Multiple Subsystems ("Intelligent Apprentice")
- Monitor/simulated control of a single subsystem - Goal and causal explanation displays - Rule-based simulation - Fault recognition / warning / limited diagnosis - Scheduling / rescheduling - Reasoning assuming standard procedures	- Coordinated control of multiple subsystems - Operator aids for unanticipated failures - Model-based simulation - Fault diagnosis for anticipated failures - Planning / replanning - Reasoning about nonstandard procedures
1993 Hierarchical Control of Multiple Subsystems ("Intelligent Assistant")	**1996 Distributed Control of Multiple Subsystems ("Intelligent Associate")**
- Multiple subsystem control: ground and space - Task oriented dialogue & human error tolerance - Fault recovery from unanticipated failures - Planning under uncertainty - Reasoning about emergency procedures	- Autonomous cooperative controllers - Goal driven natural language interface - Fault prediction and trend analysis - Automated real time planning / replanning - Reasoning / learning, supervision of on-board systems

Figure 6. AI Program Demonstration Sequence.

This figure was developed in 1985 to show how the AI research program would develop a sequence of increasingly capable demonstrations of what AI could do to control the space station. It was in keeping with the philosophy of figure 5, in which a general structure for intelligent machine behavior was to be developed. It quickly became apparent that AI program researchers had promised more than could be delivered, and they reorganized to develop and demonstrate more limited capabilities.

system for spacecraft control at JPL. The application was called SHARP (spacecraft health automated reasoning prototype).

The third task was a model-based expert system for launch processing at KSC. It was called KATE-LOX (knowledge-based autonomous test engineer–liquid oxygen). KATE is a model-based expert system shell, and KATE-LOX is an application for monitoring the loading of liquid oxygen onto the shuttle that was generated using KATE.

INCO and SHARP had an important nuance. The goal was not to just develop an expert system but to develop an entire system package that could replace the existing system that mission controllers used.

Program members vigorously discussed which of the three expert systems—SHARP, INCO, or KATE-LOX—would be the first win. The consensus, if there was one, was that the KSC application of KATE-LOX might come in first because KSC had applications people who favored AI. JPL would come in second with SHARP. The JSC INCO application was the most risky because MCC management at JSC, who would make the decision, was dead set against AI. What happened bears some discussion because a number of lessons learned emerged. It turns out that program members were wrong. INCO was the first win, SHARP came in second, and KATE-LOX is just coming in now.

INCO was the first big win. As I mentioned earlier, Muratore was the head integrated communications officer in Shuttle Mission Control at JSC. He wanted to use expert systems to help him and his team do their job. He asked his management for money and support. They said no, so he came to our research group and asked for money and support. We said yes. He told his management he had the money. He just needed their OK to develop and evaluate an alternative system. They said that he could develop his system for evaluation, but he could not touch one line of the existing code or the existing system. Most people would have been stymied—but not Muratore. He went out and bought a telemetry processor and hooked it

```
FN 226/103                          COMM MANAGEMENT                                  RR0920H  CH102

OGMT      49:15:02:31  OEMT            0:00:20:31        SITE   BDA   01    166          GN  21
RGMT      49:15:02:31  UD  RT     1     SM      M        MBFS   SPR  SM           D      BF  12
----------- S BAND PM ---------|------ KU-BAND ------------|---- NSP/COMSEC ----|------ GCIL ------
UL SS                    M     | UL  SS                   D| SELECT         2   | CONFIG CMD
STDN  SS      -204             | TDRS SEL          D      D| BIT   SYNC     BIT | KU  FS  DABL
WEST          /                | WST             /         | FRM   SYNC     FRM | PM  CMD  TV PML
EAST          /        -1131   | EST             /         | COM  SYNC      FRM | PL  CMD  FM CMC
RCVR  LCK     LK               | RF  POWER          D      | SOURCE      S      | KU  CMD
PH ERR          45             | ANT  MODE            M    | U/L    RATE     LO |------ DEU ------
COHERENT                 COH   | SEARCH              D     |        CODE     ON |  MF    DISP PWR
ANT    SEL    LLF              | DETECT             D      | D/L RATE    HI     | 1 GNC   1051  ON
   MODE       GPC              | TRACK              D      |        CODE     ON | 2 GNC    188  ON
   ELEC      1                 | KU OPER             D     | ENCR U/L   CLR  ENC | 3 GNC      1  ON
  BEAM       1                 | 216 SYNCH          D      |      D/L   CLR  CLR | 4      S      0SON
XPDR SEL                 2     | DATA GD             D     |      RCDR NSA   NSA |------ FM ------
  MODE        TDRS             | R/L HDR       TV          | RCDR  V     ON     | SS             15
PRE AMP                  2     |      LDR      OPS   RCD   | ERROR R     0.0  0.0 | MODE   ME
HEATER        1          2     | B-MODE              D     |------ CCTV --------| ANT  SEL  UP
PA          STBY     ON        |  ANGLE             D      | D/L    SEL         |     MODE    GPC
  TEMP       150       186     |----------- UHF -----------|   TEMP             | SYS  SEL   1
  RFL        0.7       1.3     | MODE S     -    -    ---  | VCU    MN          |---- SECURE TV ----
------ SPC ------ TEC ---------| 296     -78     259    -56| DOWNLINK    ENA    | ADU PWR ?
  SM            D        D     | 279     -50     243    -55| GAM   SEL   NORM   | TYPE       ?
BFS    543                     |                           | ALC                | OUTPUT    ?
---------------------- RECORDERS ---------------------------|------------------- DDM ---------------
OPS     MODE     TK    %TP      DIR      SPMTN        TEMP  | DDH   1          49:15:02:32    FR/S     100
  1      RCDA     5      45      FWD      1    RUN      102  | DDH   2          49:15:02:32    FR/S     100
  2      RCDA     2      55      REV      1    RUN      101  | DDH   3          49:15:02:32    FR/S     100
 P/L     RCDA     1      42      FWD      2    RUN      101  | DDH   4          49:15:02:32    FR/S     100
FAULT                        IMU BITE/T        3            | GPC 1234         TIME     49:15:01:55.37
```

Figure 7. A Display from the Traditional INCO *Console in the Mission Control Center at Johnson Space Center.*

The traditional computer displays in the Shuttle Mission Control Center were pages of text displaying telemetry parameters.

up at the same point at which the MCC central computer gets its telemetry. He bought a workstation. His team developed a system that processed the telemetry and then built an entirely new INCO mission control package that gave the user the capability to get the same old MCC INCO displays and controls or use new, advanced computer technology such as color computer graphic displays and rule-based expert systems. Muratore had become a *mole*, that is, a person in an operational project who is also a champion of AI. A *mole* makes technology infusion a lot easier.

Figure 7 shows one of the INCO displays, which presents a set of parameters and their values. When a controller saw a parameter that was out of tolerance or when he or she saw an alarm (usually many of them at a time), he or she could ask for a strip chart of the history of the parameter. He or she went to another room to get the chart and rolled it out on the floor to study it. He or she then went to the volumes of flowcharts and circuit diagrams and manually tried to determine what could have happened to yield the given reading.

Muratore's workstation gave color graphic presentations of the system and circuit diagrams, which changed in color from green to yellow to red as the telemetry indicated out-of-tolerance parameters. Figure 8 shows how the new system would display the type of information that is found in figure 7. It used color-coded computer graphics, and it provided rule-based fault detection and diagnosis. Interestingly enough, Muratore's system was faster than real time. His workstation showed the diagnosis more than two seconds faster than the MCC consoles could update the parameter value.

When Muratore was ready for a system test, he hooked his system up in parallel with the MCC INCO station, and a simulated mission was initiated using taped data. While this test was going on, the MCC personnel inserted a fault in the simulation that shut down the main central computer. They were testing the software that called for a switch to the backup computer. Because of a software problem, the system did not switch to the backup computer, and the MCC consoles went blank. Muratore's workstation, however, was independently hooked to the telemetry stream, and he was not affected. This test helped MCC personnel see the robustness of multiple independent workstations over a central computer. Muratore's system was simple to use, was faster than real time, and caught problems before the humans using the MCC INCO console. Eventually, John was given the OK to install his workstation at MCC alongside the traditional console. It led, after a long time, to a reduction in personnel and the development of similar applications for the other consoles at MCC. Researchers learned a lot from Muratore's experience.

INCO was the first successful application during this period of Renaissance. Then came SHARP, which did for unmanned mission control at JPL what INCO did for shuttle control at JSC. It was first applied to Voyager's telecommunications subsystem, and again, a mole was used. SHARP was used by the telecommunications team on Voyager, then it was applied to Magellan and Galileo mission control. Like INCO and the set of applications that followed INCO (called RTDS) at JSC, SHARP caught problems and either diagnosed them or allowed them to be diagnosed sooner than the human controllers working with the existing workstation.

Then KATE-LOX was tested at KSC in parallel with human controllers. It took longer to develop than INCO and SHARP because problems in KATE's model-based reasoning took some time to work out. It is now being evaluated operationally at KSC. During one flight, it predicted the failure of the Replenish Flow Meter 22 minutes before the console operators noted the failure. After this flight, the operator of the KATE-LOX system was given access to the voice network so that he could pass on such information in the future.

A number of other applications were successfully implemented that were not in the area of mission control. For example, Peter Cheeseman's work in Bayesian statistical approaches to the learning of categories in a large data set began as fundamental work but spun off as an application called AUTOCLASS. It was used to analyze the data from the infrared astronomical satellite (IRAS). The result of this study was that astronomers revised how they classified a number of sky objects.

The Renaissance period of AI at NASA was an exciting one. The program hired a group of highly competent professionals. It got the OK to "cherry pick the easy ones first." The successful applications were far less ambitious than those predicted during the Dawn era, but they legitimized the field of AI at NASA. The AI program had proven the value of its technology. The culmination of this period was the 1990 report of the Space Technology Advisory Committee that the AI program had grown in just a few years to the point where it should be emulated by the other, more established discipline research programs.

Partnerships

Winston placed his last period of AI history, the age of Partnerships, from about 1975 to 1980. It was a time when AI researchers formed ties with people from other disciplines. It was the age of the entrepreneur. NASA's age of Renaissance also evolved from one of initial successes by single, local teams to one of broader goals and partnerships with scientists and engineers from other disciplines. The initial wins such as INCO, SHARP, KATE, and AUTOCLASS were put together by a single team from a single center. Since about 1990, NASA AI researchers have formed a number of partnerships with people from other centers. Some of these partnerships are on the verge of bringing about even more revolutionary changes in the way NASA does business. Following is a description of a few of these partnerships.

Monte Zweben had been pursuing research interests in constraint-based scheduling and had developed a scheduling engine called GERRY. He found an interesting and extremely challenging scheduling problem at KSC—scheduling the ground processing of shuttle orbiters between launches. This process employs over 100 people each day for each orbiter. Zweben found that parts of this scheduling problem matched well with the characteristics of his scheduling engine, and he found some interest at KSC. Note that he also found some heavy resistance. A contractor gets paid to do the scheduling, and a contractor was getting paid to develop another scheduler. They didn't welcome Zweben's team with open arms. The team spent months learning the complexities and language of shuttle scheduling. This wasn't a straightforward process. However, Zweben found some kindred spirits among the small group of powerful high-level people who manage the flow of the orbiters through the refurbishing process. A partnership was formed among personnel from KSC, Ames Research Center (Zweben's group), Lockheed Space Operations Company, and Lockheed AI Center.

In March 1992, Wayne Bingham, the vehicle operations chief for the Orbiter *Columbia*, who is one of the powerful kindred spirits at KSC, surprised Zweben by saying that he was ready to perform an operational test of his scheduling system on the STS-50 mission. This test was almost a year ahead of the time

It was an era filled with romantic, simplistic notions of what AI would be able to do in a decade

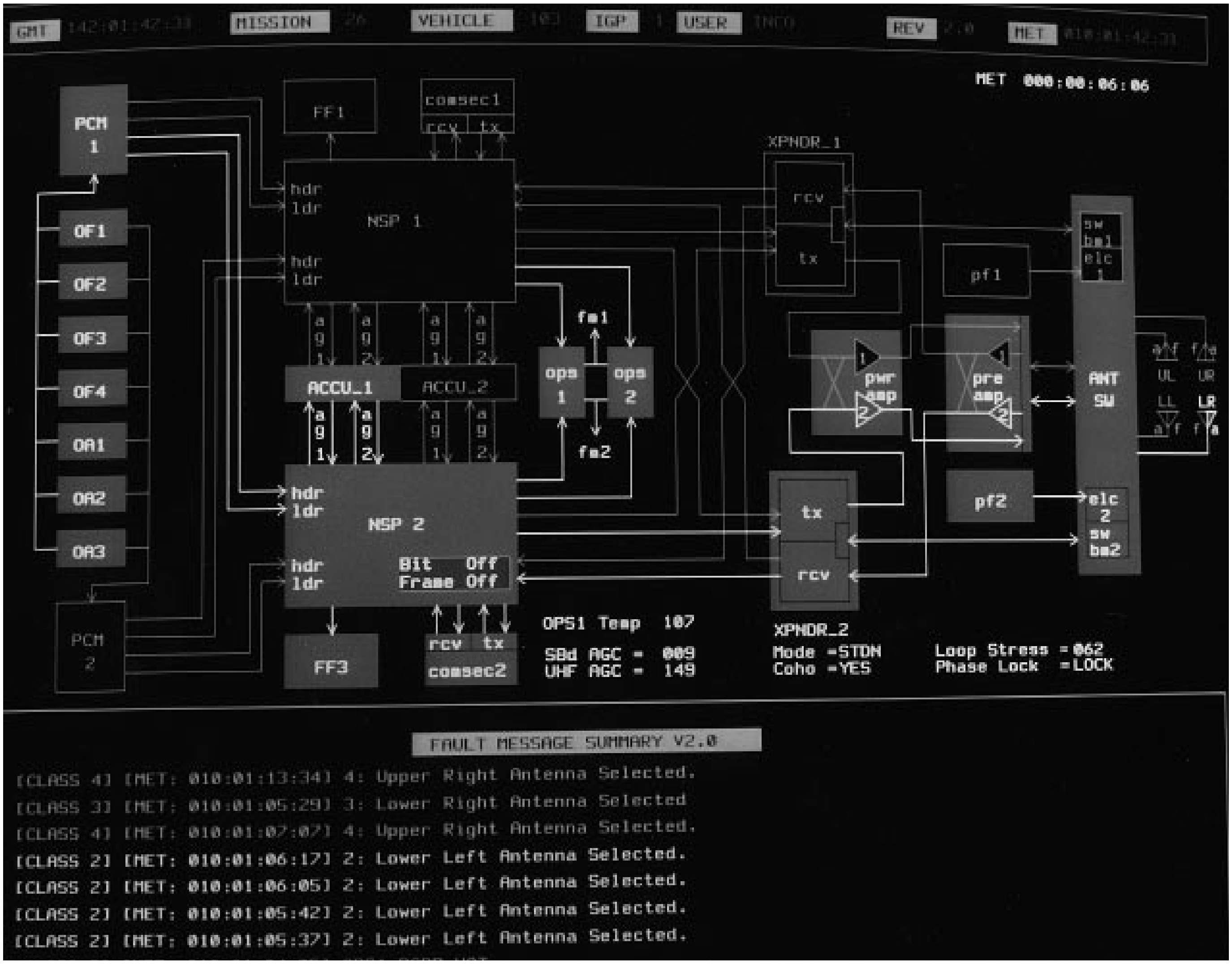


Figure 8. A New Display for the Expert System–Based INCO *Console.*

The AI program developed a rule-based INCO *system that performs monitoring and diagnosis based on the telemetry data. The display provides color graphics and fault messages. The new system provides fault identification and diagnoses before the traditional* INCO *console can update the parameters of the faulty unit.*

that Zweben's scheduler was planned to undergo an operational test of this nature. Zweben decided to go for it. This partnership just might end up saving NASA a great deal of time and money. It is estimated that the scheduler saved over $500,000 during the processing of a single orbiter.

A second entrepreneur is forming partnerships—Richard Doyle of JPL. He has been developing a method for selective monitoring of the sensors of complex systems. His approach, called SELMON, uses model-based and empirical methods for determining the importance of each sensor for monitoring the system. He has been working with the people at Marshall Space Flight Center, who have been developing Space Station *Freedom*'s environmental closed-loop life support system (ECLSS) test bed. The goal is to help them determine where to place sensors and how to combine the sensor readings for optimal monitoring of the ECLSS system.

Guy Boy, who recently left Ames to go back to France and start a cognitive science institute, had been developing an AI-based hypermedia tool for helping people navigate through complex volumes of manuals. The tool, CID (computer-integrated documentation), is now being applied to Space Station *Freedom*'s Program Requirement Document to help people use the document more easily.

Mark Drummond, who has been working on an integrated approach to planning, scheduling, and control, has formed a partnership with the makers of an automatic telescope to improve the science return from such telescopes.

Amy Lansky has formed a partnership with the Center for Space Construction in Colorado to attempt to apply GEMPLAN, her tool for multiagent planning, to space-construction tasks.

Dave Thompson and Rich Levinson have been working with Rocco Mancenelli, the developer of a new instrument called DTA-GC, to automate the instrument. They have automated the analysis of the output of the instrument and have developed a method for planning the next experimental protocol. I believe this instrument is a forerunner of a class of intelligent instruments that will be finding their way to the Moon and Mars.

These are a sample of the partnerships that characterize the current era in the evolution of NASA's AI research program. Thus, our journey through the rites of passage of NASA's AI research program is completed.

Lessons Learned

It can be seen from the last section of this article that Pat Winston's analysis of the history of AI was predictive of the stages that NASA's AI program went through. This fact alone indicates that valuable lessons in program management and development can be learned from the experience of others, which leads to the first of our lessons learned:

> **Santayana's Statement:** Those ignorant of history are doomed to repeat it.

The implication in this statement is that anyone who is planning to start or manage an AI program would do well to try to ascertain what he or she can from the experience of DARPA, NASA, the National Science Foundation, the European Space Agency, and so on. Unfortunately, not much is written down. One of these lessons from history is as follows:

> **Sitting Bull's First Law of Teepee Design:** You need at least three poles to build a teepee.

> **Program Management Corollary to Sitting Bull's Law:** To successfully carry out a research program, you need three things: personnel, funding, and facilities.

> **The Universal Dilemma to the Corollary:** You can't get the people and the facilities without the money, and they don't want to give you the money unless you have the people and the facilities.

This process is a little like trying to get a loan. If you need it, you can't qualify. If you don't need it, you can qualify. At NASA, what the AI program was missing was a cadre of competent personnel, a place to house them, and money. First, the program got the funding. Later, the program got the people and housed many of them in trailers. Finally, it got nice facilities.

It would be wonderful if a program only had to be sold once. Unfortunately, this is rarely the case. I was given an important piece of advice by a man named George Deutsch back in September 1979. However, because it is about *Stayin' Alive*, I named it after John Travolta.

> **Travolta's Theorem:** If you want your program to survive any length of time, it must have two components: a strong fundamental research program and a component that regularly turns out things that can be used.

George Deutsch said that as management changes occur, the pendulum swings between pressures toward fundamental work and applied work. Having some of both enhances the chances of longevity.

The laws listed earlier are concerned with selling the program both initially and continually. The laws that follow are heuristics for success in carrying out an AI research and development program. The most important law concerning getting a successful applications program going comes straight from Ed Feigenbaum. We saw it earlier in the article.

> **Ed's Edict:** Cherry pick the easy ones first.

For some reason, there is pressure to try the hard ones first. This pressure must be resisted.

The second law of successful applications holds equally well in the AI community and in the CIA, but it does not apply to gardening:

> **J. Edgar Hoover's Formula:** Whenever possible, use moles.

The first successful application at NASA, which I discussed earlier, was INCO in Shuttle Mission Control. This effort was led by the head integrated communications officer. He was both one of us and one of them. What a combination! The insurrection came from within. The first win at JPL, which was the SHARP application to Voyager control, was done using one of the Voyager controllers. Nothing works better.

Muratore has turned out to be a powerful force for AI in NASA. He has developed a number of laws based on his INCO experience. The most important is as follows:

Muratore's Motto: Technology transfer is a body contact sport.

Muratore is a great fan of having the technologists get right down in the trenches early and staying there until its over. This approach works.

Remember that MCC wouldn't let Muratore touch a line of its code, so his team developed its own mission control system that could operate right along the standard system. Thus, we have the best method of showing that the new approach should be adopted:

The John Henry Test: Demonstrate the new way right next to the old way right in the operational environment.

This approach worked at Shuttle Mission Control, Satellite Mission Control at JPL, and Launch Control at KSC. There is no other way.

The point at which AI at NASA went from the Dark Ages to the Renaissance is the point at which we realized the following:

The Maximization Maxim: Don't maximize the amount of AI in an application. Maximize the system's utility to the user.

Muratore used old AI technology and not much of it at first, but the overall new system was obviously better than the existing one. After the users decided they liked the system, it was relatively easy to add more AI. However, AI was not the first or second thing you noticed about Muratore's new system. The first thing you noticed was the color graphics and the modern workstation. The second thing you noticed was how much easier it made your job.

Convincing potential users that you can develop an application that will be useful to them is only the first battle. You need more of the experts' precious time than they thought they signed up for. Zweben's development of a scheduler for shuttle launch processing is a prime example. The human schedulers and their supervisors are busy people, and they don't need distractions, which brings us to the following law:

Gypsy Rose Lee's Law of Keeping Their Interest: To keep a customer's interest long enough to get the job done, you need to give them a little along the way.

Zweben couldn't keep the attention of the flow manager for two years while the full blown scheduler was developed and validated. The answer was to plan the project so that along the way, tools were produced for the user, and services were provided to the user.

In trying to find friends who can help increase the chances of success of a planned AI application, never forget what Jim Croce sang:

Jim Croce's Exhortation: You don't tug at superman's cape, you don't spit into the wind, you don't pull the mask off the old lone ranger, and you probably won't be able to make friends with the local management information system manager.

This exhortation is especially true if you are trying to replace his software and his hardware. When you run into this problem, you need to find a 2000-pound gorilla, like his or her boss—someone who has a bigger picture. By the way, Zweben learned this lesson well. Not everyone welcomed his team at KSC. Some of those with vested interests were not interested in the system he proposed. The thing that turned the tide was when two of the shuttle flow managers, who are quintessential 2000-pound gorillas at NASA, put their confidence in him and gave him their vocal and public support in word and deed.

NASA's AI program started with a single large applications project, and the milestones it had were based on the development of new technology. This approach did not work. In an organization such as NASA, it is unlikely that a research program would enjoy longevity unless it results in something tangible to the agency in a reasonable length of time. Thus, we have this paradigm:

The Pre-Poultry Paradigm: Don't put all your eggs in one application, and don't bet on which one will work out first or best.

NASA's AI program gave up its reliance on a large, single demonstration program and replaced it with the INCO, SHARP, and KATE-LOX applications. Each of the projects these applications were being developed for went through unpredictable circumstances that modified the timetable and the nature of the applications that were developed, giving rise to the paradigm's lemma:

Pre-Poultry Paradigm Lemma: Plan for serendipity.

The NASA AI program has so often profited from, or been hindered by, unpredictable events that researchers now build time for such events into their planning, not only for applications but also for basic research.

With regard to basic research, the AI program has come upon a finding here as well. The program has fundamental research going on in planning, scheduling, learning, reasoning about complex physical systems, and other areas. Friedland, as leader of the largest group of researchers, has instituted his own principle:

> **Peter's Principle:** All the research we sponsor must be done in domains of interest to the agency.

Although the research might be basic, this principle has had a number of beneficial effects. It causes researchers to form partnerships with project people to get access to real data. This approach makes it easier to convince NASA headquarters that the research is relevant because it has face validity. Also, it keeps researchers from working on toy problems. It has also led to some unplanned but wonderful applications, such as Zweben's scheduler at KSC and Cheeseman's AUTOCLASS tool.

No research or development would be possible in our world without proposals. I have learned a lot about the process of proposal generation and review. From this learning, I have developed my own motto:

> **Mel's Motto:** Fund good people, not good words.

I believe that the biggest reason for the success of this program so far has been the quality of the people. They are not only creative, they also work hard. This ethic is personified by Zweben, for whom is named this method:

> **Zweben's Method:** Go for it. Sleep next week.

Conclusion

I described the history of AI at NASA and delineated some of the lessons learned. One of the problems with this approach is that one can come up with blinding flashes of the obvious. It is like the 12-year-old boy who discovered during a grammar lesson that all throughout his or her life, he or she had been speaking in paragraphs. Some of the laws I enumerated are truisms. I just wish that everyone had been aware of them when the AI program was started. I hope they will be of use to you.

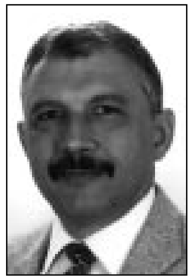
References

JPL. 1980. Machine Intelligence and Robotics: Report of the NASA Study Group, Report 715-32, Jet Propulsion Laboratory, Pasadena, California.

Montemerlo, M. D., and Holcomb, L. B. 1986. NASA's Automation and Robotics Technology Development Program: A White Paper, Office of Aeronautics and Space Technology, NASA Headquarters, Washington, D.C.

NASA. 1988. Space Station Advanced Automation Study Final Report, Office of Space Station, NASA Headquarters, Washington, D.C.

Winston, P. H. 1984. Perspective. In *The AI Business*, eds. P. H. Winston and K. A. Prendergast, 1–13. Cambridge, Mass.: MIT Press.



Mel Montemerlo is the manager of NASA's Artificial Intelligence Research and Development Program. He received a Ph.D. in educational psychology from Pennsylvania State University in 1970, where he specialized in human learning. After nine years in the field of human factors, he joined NASA Headquarters to manage its Aeronautical Human Factors Research Program. In 1983, he instituted a Space Human Factors Research Program. He initiated AI research in each of these programs. He has been the manager of the Artificial Intelligence Research and Development Program since it was started in 1985. The goal of the program is to develop AI technology applicable to NASA's needs and to transfer it to operational use.